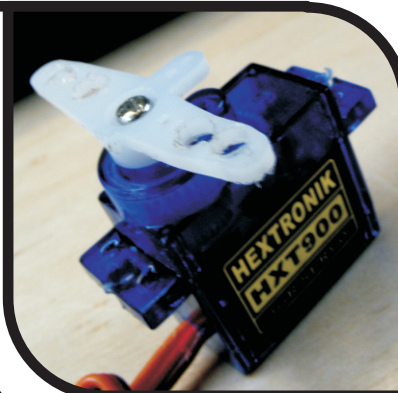


# ..A Single Servo.. ..Servos..



## WHAT WE'RE DOING:

Spinning a motor is good fun but when it comes to projects where motion control is required they tend to leave us wanting more. The answer? Hobby servos. They are mass produced, widely available and cost anything from a couple of dollars to hundreds. Inside is a small gearbox (to make the movement more powerful) and some electronics (to make it easier to control). A standard servo is positionable from 0 to 180 degrees. Positioning is controlled through a timed pulse, between 1.25 milliseconds (0 degrees) and 1.75 milliseconds (180 degrees) (1.5 milliseconds for 90 degrees). Timing varies between manufacturer. If the pulse is sent every 25-50 milliseconds the servo will run smoothly. One of the great features of the Arduino is it has a software library that allows you to control two servos (connected to pin 9 or 10) using a single line of code.

## THE CIRCUIT:

### Parts:



**CIRC-04  
Breadboard Sheet  
x1**



**2 Pin Header  
x4**



**3 Pin Header  
x1**

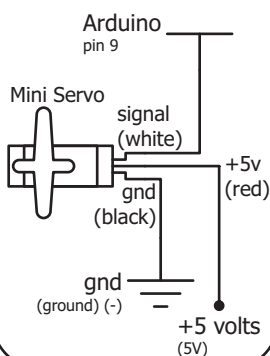


**Wire**



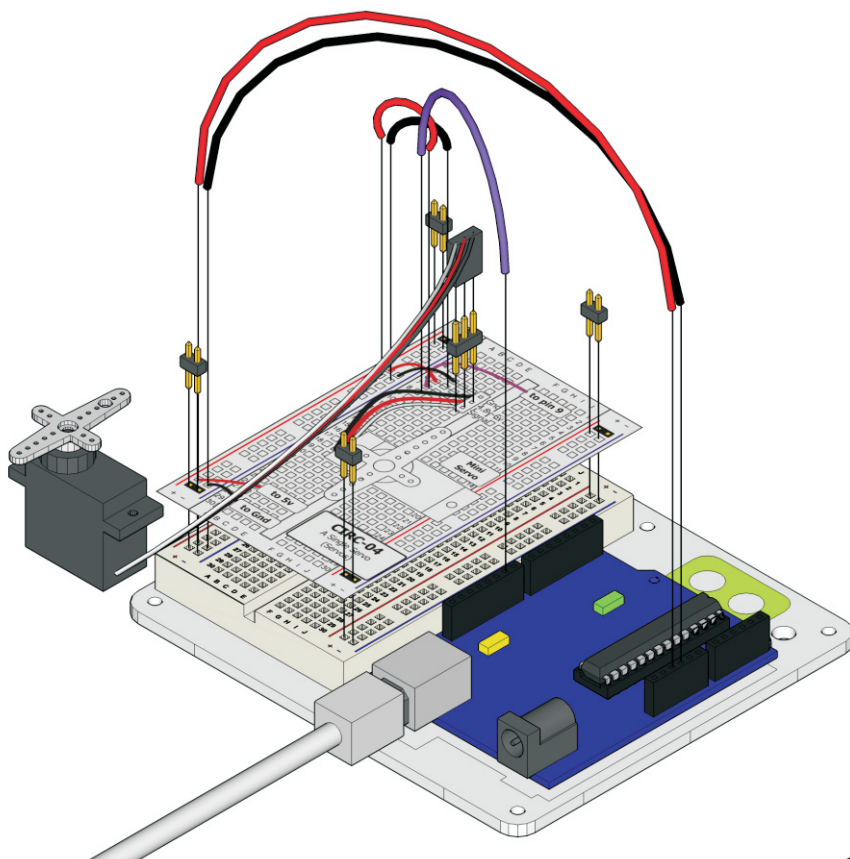
**Mini Servo  
x1**

### Schematic



### The Internet

..download..  
breadboard layout sheet  
<http://ardx.org/BBLS04>  
..view..  
assembly video  
<http://ardx.org/VIDE04>



**CODE** (no need to type everything in just click)**File > Examples > Servo > Sweep**

(example from the great arduino.cc site, check it out for other great ideas)

```
// Sweep
// by BARRAGAN <http://barraganstudio.com>

#include <Servo.h>
Servo myservo;  // create servo object to control a servo
int pos = 0;    // variable to store the servo position

void setup() {
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
}

void loop() {
  for(pos = 0; pos < 180; pos += 1)  // goes from 0 degrees to 180 degrees
  {                                  // in steps of 1 degree
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15ms for the servo to reach the position
  }
  for(pos = 180; pos>=1; pos--=1)    // goes from 180 degrees to 0 degrees
  {
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15ms for the servo to reach the position
  }
}
```

**NOT WORKING?** (3 things to try)**Servo Not Twisting?**

Even with colored wires it is still shockingly easy to plug a servo in backwards. This might be the case.

**Still Not Working**

A mistake we made a time or two was simply forgetting to connect the power (red and brown wires) to +5 volts and ground.

**Fits and Starts**

If the servo begins moving then twitches, and there's a flashing light on your Arduino board, the power supply you are using is not quite up to the challenge. Using a fresh battery instead of USB should solve this problem.

**MAKING IT BETTER****Potentiometer control:**

We have yet to experiment with inputs but if you would like to read ahead, there is an example program **File > Servo >**

**Knob**. This uses a potentiometer (CIRC08) to control the servo. You can find instructions online here:

<http://ardx.org/KNOB>

**Self timing:**

While it is easy to control a servo using the Arduino's included library sometimes it is fun to figure out how to program something yourself. Try it. We're controlling the pulse directly so you could use this method to control servos on any of the Arduino's 20 available pins (you need to highly optimize this code before doing that).

```
int servoPin = 9;

void setup(){
  pinMode(servoPin,OUTPUT);
}
```

```
void loop() {
  int pulseTime = 2100; //(the number of microseconds
                        //to pause for (1500 90 degrees
                        // 900 0 degrees 2100 180 degrees)

  digitalWrite(servoPin, HIGH);
  delayMicroseconds(pulseTime);
  digitalWrite(servoPin, LOW);
  delay(25);
}
```

**Great ideas:**

Servos can be used to do all sorts of great things, here are a few of our favorites.

Xmas Hit Counter  
<http://ardx.org/XMAS>

Open Source Robotic Arm (uses a servo controller as well as the Arduino)  
<http://ardx.org/RARM>

Servo Walker  
<http://ardx.org/SEWA>

**MORE, MORE, MORE:**

More details, where to buy more parts, where to ask more questions:

**<http://ardx.org/CIRC04>**